

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2002年 9月14日

出願番号
Application Number: 特願2002-307637

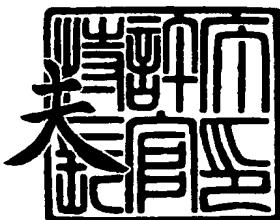
[ST. 10/C]: [JP2002-307637]

出願人
Applicant(s): 株式会社リコー

2003年 8月11日

特許庁長官
Commissioner,
Japan Patent Office

今井康



【書類名】

特許願

【整理番号】

0203932

【提出日】

平成14年 9月14日

【あて先】

特許庁長官殿

【国際特許分類】

G03G 21/00 370

【発明の名称】

画像形成装置およびアプリ情報取得方法

【請求項の数】

19

【発明者】

【住所又は居所】 東京都大田区中馬込1丁目3番6号 株式会社リコー内

【氏名】 秋吉 邦洋

【発明者】

【住所又は居所】 東京都大田区中馬込1丁目3番6号 株式会社リコー内

【氏名】 田中 浩行

【特許出願人】

【識別番号】 000006747

【氏名又は名称】 株式会社リコー

【代理人】

【識別番号】 100089118

【弁理士】

【氏名又は名称】 酒井 宏明

【手数料の表示】

【予納台帳番号】 036711

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9808514

【書類名】 明細書

【発明の名称】 画像形成装置およびアプリ情報取得方法

【特許請求の範囲】

【請求項 1】 画像形成処理にかかるユーザサービスを提供する画像形成装置であって、

前記画像形成装置にコピーされたアプリケーションを起動可能な状態にするインストール処理を行う前に、インストールが可能なアプリケーションか否かを判断するためのアプリケーションに関するアプリ情報の取得を行うアプリ情報取得手段

を備えたことを特徴とする画像形成装置。

【請求項 2】 画像形成処理で使用されるハードウェア資源と、
画像形成処理にかかるユーザサービスにそれぞれ固有の処理を行うアプリケーションと、

前記アプリケーションと前記ハードウェア資源との間に介在し、ユーザサービスを提供する際に、アプリケーションの少なくとも2つが共通的に必要とする前記ハードウェア資源の獲得要求、管理、実行制御並びに画像形成処理を行うコントロールサービスと、

をさらに備えたことを特徴とする請求項1に記載の画像形成装置。

【請求項 3】 前記アプリ情報取得手段は、前記画像形成装置にコピーされたインストール処理前のアプリケーションが自ら保有するアプリ情報を取得することにより行うことを特徴とする請求項1または2に記載の画像形成装置。

【請求項 4】 前記アプリ情報取得手段は、前記画像形成装置にコピーされたアプリケーションインストール処理前のアプリケーションを仮起動させることにより、当該アプリケーションのアプリ情報を取得することを特徴とする請求項1～3のいずれか一つに記載の画像形成装置。

【請求項 5】 前記アプリ情報取得手段は、ネットワーク経由で前記画像形成装置にコピーされたインストール処理前のアプリケーションのアプリ情報を取得することを特徴とする請求項1または2に記載の画像形成装置。

【請求項 6】 前記アプリ情報取得手段が取得したアプリ情報を記憶する記

憶手段をさらに備えたことを特徴とする請求項1～5のいずれか一つに記載の画像形成装置。

【請求項7】 前記記憶手段は、ハードディスク、不揮発性メモリ、および各種カードの少なくとも一つであって、

前記アプリ情報取得手段は、それらの記憶媒体に記憶された情報を読み出すことによりアプリ情報を取得することを特徴とする請求項6に記載の画像形成装置。

【請求項8】 前記アプリ情報取得手段が取得するアプリ情報は、アプリケーションごとの固有の情報が含まれたプロダクトIDであることを特徴とする請求項1～7のいずれか一つに記載の画像形成装置。

【請求項9】 前記プロダクトIDには、対応するアプリケーションのベンダー名とアプリケーション名に加えて、バージョン情報と使用リソース情報の少なくとも一方が含まれていることを特徴とする請求項8に記載の画像形成装置。

【請求項10】 前記記憶手段は、インストール可能なアプリケーションのプロダクトIDを記憶し、判断対象となるアプリケーションのプロダクトIDがその記憶手段に記憶されているか否かによって、インストール可能なアプリケーションか否かを自動的に判断することを特徴とする請求項8または9に記載の画像形成装置。

【請求項11】 画像形成処理にかかるユーザサービスを提供する画像形成装置上で、アプリケーションに関するアプリ情報を取得するアプリ情報取得方法であって、

前記画像形成装置にコピーされたアプリケーションを起動可能な状態にするインストール処理を行う前に、インストールが可能なアプリケーションか否かを判断するためのアプリケーションに関するアプリ情報を取得するアプリ情報取得ステップ

を含むことを特徴とするアプリ情報取得方法。

【請求項12】 前記アプリ情報取得ステップは、前記画像形成装置にコピーされたインストール処理前のアプリケーションが自ら保有するアプリ情報を取得することにより行うことを特徴とする請求項11に記載のアプリ情報取得方法

【請求項13】 前記アプリ情報取得ステップは、前記画像形成装置にコピーされたアプリケーションインストール処理前のアプリケーションを仮起動させることにより、当該アプリケーションのアプリ情報を取得することを特徴とする請求項11または12に記載のアプリ情報取得方法。

【請求項14】 前記アプリ情報取得ステップは、ネットワーク経由で前記画像形成装置にコピーされたインストール処理前のアプリケーションのアプリ情報を取得することを特徴とする請求項11に記載のアプリ情報取得方法。

【請求項15】 前記アプリ情報取得ステップによって取得したアプリ情報を記憶する記憶ステップをさらに含むことを特徴とする請求項11～14のいずれか一つに記載のアプリ情報取得方法。

【請求項16】 前記アプリ情報取得ステップは、前記記憶ステップによりハードディスク、不揮発性メモリ、および各種カードの少なくとも一つに記憶された情報を読み出すことによってアプリ情報を取得することを特徴とする請求項15に記載のアプリ情報取得方法。

【請求項17】 前記アプリ情報取得ステップによって取得するアプリ情報は、アプリケーションごとの固有の情報が含まれたプロダクトIDであることを特徴とする請求項11～16のいずれか一つに記載のアプリ情報取得方法。

【請求項18】 前記プロダクトIDには、対応するアプリケーションのベンダー名とアプリケーション名に加えて、バージョン情報と使用リソース情報の少なくとも一方が含まれていることを特徴とする請求項17に記載のアプリ情報取得方法。

【請求項19】 前記記憶ステップは、インストールを可能とするアプリケーションのプロダクトIDを記憶し、判断対象となるアプリケーションのプロダクトIDがその記憶ステップに記憶されているか否かによって、インストール可能なアプリケーションか否かを自動的に判断することを特徴とする請求項17または18に記載のアプリ情報取得方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、コピー、プリンタ、スキャナおよびファクシミリなどの画像形成処理にかかるユーザサービスを提供する画像形成装置で、インストール処理を行うアプリケーションに関するアプリ情報取得する画像形成装置およびアプリ情報取得方法に関するものである。

【0002】**【従来の技術】**

近年では、プリンタ、コピー、ファクシミリ、スキャナなどの各装置の機能を1つの筐体内に収納した画像形成装置（以下、「複合機」という。）が知られている。この複合機は、1つの筐体内に表示部、印刷部および撮像部などを設けるとともに、プリンタ、コピーおよびファクシミリ装置にそれぞれ対応した3種類のソフトウェアを設け、これらのソフトウェアを切り替えることによって、当該装置をプリンタ、コピー、スキャナまたはファクシミリ装置として動作させるものである。

【0003】

このような従来の複合機では、プリンタ、コピー、ファクシミリ、スキャナなどの各機能単位ですべてのアプリケーションプログラムが起動される。これら既存のアプリケーションの場合、各アプリケーションは、起動可能な状態にするためのインストール処理が事前に行われている。すなわち、従来の複合機では、あらかじめ搭載されている既存のアプリケーションの場合、起動可能な状態にするインストール処理を行うにあたって、アプリケーションと複合機とのバージョンの整合性や使用するリソース量などが事前にわかっているため、通常にインストール処理を行って、起動させて実行するだけでよく、インストール前にバージョンやリソース量などをチェックする必要はなかった。

【0004】**【発明が解決しようとする課題】**

ところで、このような従来の複合機では、プリンタ、コピー、スキャナおよびファクシミリ装置に対応するソフトウェアをそれぞれ別個に設けているため、各ソフトウェアの開発に多大の時間を要する。このため、出願人は、表示部、印刷

部および撮像部などの画像形成処理で使用されるハードウェア資源を有し、プリンタ、コピーまたはファクシミリなどの各ユーザサービスにそれぞれ固有の処理を行うアプリケーションを複数搭載し、これらのアプリケーションとハードウェア資源との間に介在して、ユーザサービスを提供する際に、アプリケーションの少なくとも2つが共通的に必要とするハードウェア資源の管理、実行制御並びに画像形成処理を行う各種コントロールサービスからなるプラットホームを備えた画像形成装置（複合機）を発明した。

【0005】

このような新規な複合機では、アプリケーションの少なくとも2つが共通的に必要とするサービスを提供するコントロールサービスをアプリケーションと別個に設けた構成となっているため、複合機の出荷後にユーザもしくは第三者であるサードベンダが新規なアプリケーションを開発して、複合機に搭載可能な構成となっている。このため、かかる複合機では、出荷時に搭載されているコピー、プリンタ、スキャナ、ファクシミリといった既存のアプリケーション以外に、従来の複合機とは異なるユーザやサードベンダが開発した新規アプリケーションなどを複合機にコピーした後、そのアプリケーションを起動可能な状態にするためのインストール処理する必要がある。上記した複合機の既存のアプリケーションの場合は、バージョンの整合性や使用リソース量があらかじめわかっているため、インストールして起動させたとしても、バージョンの不整合やリソース不足による動作の不具合が生じるおそれはない。しかしながら、サードベンダなどが開発する新規アプリケーションの場合は、搭載する複合機の状況（バージョンやリソース容量）まで考慮して開発していないため、その新規アプリケーションを起動させた場合に複合機の動作が不安定になる可能性がある。そこで、とりあえずインストールしておいて、起動までにチェックすれば動作が不安定になることが防止できるとも考えられる。しかし、起動させると複合機の動作が不安定になる可能性のあるアプリケーションをインストールしておくのは無意味なことである。このように、新規アプリケーションが搭載可能な新規な複合機では、新規アプリケーションのバージョンチェックやリソースチェックを行わずに起動させると、動作が不安定となったり、動作が途中で停止するなど従来の複合機では問題にな

らなかった新規な課題が生じてくる。

【0006】

この発明は上記に鑑みてなされたもので、新規アプリケーションを複合機に搭載して、起動可能な状態とするインストール処理の前に、インストールが可能なアプリケーションか否かを判断するために必要なアプリケーションに関する正確なアプリ情報が取得できる画像形成装置およびアプリ情報取得方法を得ることを目的とする。

【0007】

【課題を解決するための手段】

上記目的を達成するため、請求項1にかかる発明は、画像形成処理にかかるユーザサービスを提供する画像形成装置であって、前記画像形成装置にコピーされたアプリケーションを起動可能な状態にするインストール処理を行う前に、インストールが可能なアプリケーションか否かを判断するためのアプリケーションに関するアプリ情報の取得を行うアプリ情報取得手段を備えたことを特徴とする。

【0008】

この請求項1にかかる発明によれば、画像形成装置にコピーされたアプリケーションを起動可能な状態にするインストール処理を行う前に、アプリ情報取得手段によって、インストールが可能なアプリケーションか否かを判断するためのアプリケーションに関するアプリ情報の取得を行うようにしたことで、インストール前でもインストールが可能なアプリケーションか否かを正確な判断を下すことができる。

【0009】

また、請求項2にかかる発明は、請求項1に記載の画像形成装置において、画像形成処理で使用されるハードウェア資源と、画像形成処理にかかるユーザサービスにそれぞれ固有の処理を行うアプリケーションと、前記アプリケーションと前記ハードウェア資源との間に介在し、ユーザサービスを提供する際に、アプリケーションの少なくとも2つが共通的に必要とする前記ハードウェア資源の獲得要求、管理、実行制御並びに画像形成処理を行うコントロールサービスとをさらに備えたことを特徴とする。

【0010】

この請求項2にかかる発明によれば、画像形成処理で使用されるハードウェア資源と、画像形成処理にかかるユーザサービスにそれぞれ固有の処理を行うアプリケーションと、前記アプリケーションと前記ハードウェア資源との間に介在し、ユーザサービスを提供する際に、アプリケーションの少なくとも2つが共通的に必要とするハードウェア資源の獲得要求、管理、実行制御並びに画像形成処理を行うコントロールサービスとをさらに備えていることで、複数のアプリケーションを作成する際に、共通化できる部分をコントロールサービス（プラットホーム）として共通化することより、それ以外の部分を作成すればよいので、アプリケーションの開発が容易となり、かつ、短期間で開発することができる。

【0011】

また、請求項3にかかる発明は、請求項1または2に記載の画像形成装置において、前記アプリ情報取得手段は、前記画像形成装置にコピーされたインストール処理前のアプリケーションが自ら保有するアプリ情報を取得することによりを行うことを特徴とする。

【0012】

この請求項3にかかる発明によれば、画像形成装置にコピーされたインストール処理前のアプリケーションが自らアプリ情報を保有している場合、アプリ情報取得手段は、このアプリが保有するアプリ情報を取得するようにしたため、アプリケーションから直接正確な情報を取得することができる。

【0013】

また、請求項4にかかる発明は、請求項1～3のいずれか一つに記載の画像形成装置において、前記アプリ情報取得手段は、前記画像形成装置にコピーされたアプリケーションインストール処理前のアプリケーションを仮起動させることにより、当該アプリケーションのアプリ情報を取得することを特徴とする。

【0014】

この請求項4にかかる発明によれば、画像形成装置にコピーされたアプリケーションインストール処理前のアプリケーションを仮起動することで、アプリ情報取得手段は、アプリケーションのアプリ情報を取得するようにしたため、通常起

動前のアプリケーションに対してもプロセス間通信を用いて情報をやり取りすることにより、正確なアプリ情報を取得することができる。

【0015】

また、請求項5にかかる発明は、請求項1または2に記載の画像形成装置において、前記アプリ情報取得手段は、ネットワーク経由で前記画像形成装置にコピーされたインストール処理前のアプリケーションのアプリ情報を取得することを特徴とする。

【0016】

この請求項5にかかる発明によれば、アプリ情報取得手段は、ネットワーク経由で画像形成装置にコピーされたインストール処理前のアプリケーションのアプリ情報を取得するようにしたため、常に最新の情報と更新可能なネットワークデータを使って正確なアプリ情報を取得することができる。

【0017】

また、請求項6にかかる発明は、請求項1～5のいずれか一つに記載の画像形成装置において、前記アプリ情報取得手段が取得したアプリ情報を記憶する記憶手段をさらに備えたことを特徴とする。

【0018】

この請求項6にかかる発明によれば、アプリ情報取得手段が取得したアプリ情報を記憶手段に記憶するようにしたため、一旦取得したアプリ情報を必要に応じていつでも取り出せることから、データベースのように使用することが可能となり、アプリ情報取得に要する時間を短縮して、処理の迅速化が図れる。

【0019】

また、請求項7にかかる発明は、請求項6に記載の画像形成装置において、前記記憶手段は、ハードディスク、不揮発性メモリ、および各種カードの少なくとも一つであって、前記アプリ情報取得手段は、それらの記憶媒体に記憶された情報を読み出すことによりアプリ情報を取得することを特徴とする。

【0020】

この請求項7にかかる発明によれば、ハードディスク、不揮発性メモリ、および各種カードの少なくとも一つから記憶手段が構成され、アプリ情報取得手段に

よって、それらの記憶媒体に記憶された情報を読み出すことでアプリ情報を取得するようにしたため、種々の記憶媒体をアプリ情報の記憶手段として利用するこ^トが可能となり、状況に応じて記憶媒体を使い分けることができる。

【0021】

また、請求項8にかかる発明は、請求項1～7のいずれか一つに記載の画像形成装置において、前記アプリ情報取得手段が取得するアプリ情報は、アプリケーションごとの固有の情報が含まれたプロダクトIDであることを特徴とする。

【0022】

この請求項8にかかる発明によれば、アプリ情報取得手段が取得するアプリ情報として、アプリケーションごとの固有の情報を含むプロダクトIDを用いたため、アプリ情報のやり取りをこのプロダクトIDだけで、簡易かつ迅速に行うことができるとともに、アプリ情報が一元管理できることから、情報の信頼性を向上させることができる。

【0023】

また、請求項9にかかる発明は、請求項8に記載の画像形成装置において、前記プロダクトIDには、対応するアプリケーションのベンダー名とアプリケーション名に加えて、バージョン情報と使用リソース情報の少なくとも一方が含まれていることを特徴とする。

【0024】

この請求項9にかかる発明によれば、プロダクトIDには、対応するアプリケーションのベンダー名、アプリケーション名、および、バージョン情報と使用リソース情報の少なくとも一方が含まれているため、アプリ情報としてプロダクトIDを取得するだけで、アプリケーションの起動時に問題となるバージョンの整合性やリソース不足による動作の不安定要因の有無を正確に判断することができる。

【0025】

また、請求項10にかかる発明は、請求項8または9に記載の画像形成装置において、前記記憶手段は、インストール可能なアプリケーションのプロダクトIDを記憶し、判断対象となるアプリケーションのプロダクトIDがその記憶手

段に記憶されているか否かによって、インストール可能なアプリケーションか否かを自動的に判断することを特徴とする。

【0026】

この請求項10にかかる発明によれば、記憶手段には、インストール可能なアプリケーションのプロダクトIDを記憶するようにし、判断対象のアプリケーションのプロダクトIDがその記憶手段に記憶されているか否かを調べるだけで、そのアプリケーションがインストール可能か否かを自動的に判断できるようにしたため、簡易な構成だけで迅速かつ正確にインストール可能なアプリケーションか否かを自動判断することができる。

【0027】

また、請求項11にかかる発明は、画像形成処理にかかるユーザサービスを提供する画像形成装置上で、アプリケーションに関するアプリ情報を取得するアプリ情報取得方法であって、画像形成装置にコピーされたアプリケーションを起動可能な状態にするインストール処理を行う前に、インストールが可能なアプリケーションか否かを判断するためのアプリケーションに関するアプリ情報を取得するアプリ情報取得ステップを含むことを特徴とする。

【0028】

この請求項11にかかる発明によれば、画像形成装置にコピーされたアプリケーションを起動可能な状態にするインストール処理を行う前に、アプリ情報を取得ステップによって、インストールが可能なアプリケーションか否かを判断するためのアプリケーションに関するアプリ情報を取得を行うようにしたため、インストール前でもインストールが可能なアプリケーションか否かを正確な判断を下すことができる。

【0029】

また、請求項12にかかる発明は、請求項11に記載のアプリ情報を取得方法について、前記アプリ情報を取得ステップは、前記画像形成装置にコピーされたインストール処理前のアプリケーションが自ら保有するアプリ情報を取得することによりを行うことを特徴とする。

【0030】

この請求項12にかかる発明によれば、画像形成装置にコピーされたインストール処理前のアプリケーションが自らアプリ情報を保有している場合、アプリ情報取得ステップは、このアプリケーションが保有するアプリ情報を取得するようとしたため、アプリケーションから直接正確な情報を取得することができる。

【0031】

また、請求項13にかかる発明は、請求項11または12に記載のアプリ情報取得方法において、前記アプリ情報取得ステップは、前記画像形成装置にコピーされたアプリケーションインストール処理前のアプリケーションを仮起動させることにより、当該アプリケーションのアプリ情報を取得することを特徴とする。

【0032】

この請求項13にかかる発明によれば、画像形成装置にコピーされたアプリケーションインストール処理前のアプリケーションを仮起動することで、アプリ情報取得ステップによって、アプリケーションのアプリ情報を取得するようにしたため、通常起動前のアプリケーションに対してもプロセス間通信を用いて情報をやり取りすることが可能となり、正確なアプリ情報を取得することができる。

【0033】

また、請求項14にかかる発明は、請求項11に記載のアプリ情報取得方法において、前記アプリ情報取得ステップは、ネットワーク経由で前記画像形成装置にコピーされたインストール処理前のアプリケーションのアプリ情報を取得することを特徴とする。

【0034】

この請求項14にかかる発明によれば、アプリ情報取得ステップは、ネットワーク経由で画像形成装置にコピーされたインストール処理前のアプリケーションのアプリ情報を取得するようにしたため、常に最新の情報と更新可能なネットワークデータを使って正確なアプリ情報を取得することができる。

【0035】

また、請求項15にかかる発明は、請求項11～14のいずれか一つに記載のアプリ情報取得方法において、前記アプリ情報取得ステップによって取得したアプリ情報を記憶する記憶ステップをさらに含むことを特徴とする。

【0036】

この請求項15にかかる発明によれば、アプリ情報取得ステップによって取得したアプリ情報を記憶ステップにて記憶するようにしたため、一旦取得したアプリ情報を必要に応じていつでも取り出すことができ、データベースのような使用が可能となり、アプリ情報取得に要する時間を短縮して、処理の迅速化が図れる。

【0037】

また、請求項16にかかる発明は、請求項15に記載のアプリ情報取得方法において、前記アプリ情報取得ステップは、前記記憶ステップによりハードディスク、不揮発性メモリ、および各種カードの少なくとも一つに記憶された情報を読み出すことによってアプリ情報を取得することを特徴とする。

【0038】

この請求項16にかかる発明によれば、アプリ情報取得ステップは、記憶ステップによってハードディスク、不揮発性メモリ、および各種カードの少なくとも一つの記憶媒体に記憶された情報を読み出すことでアプリ情報が取得できるようにしたため、アプリ情報を記憶しておく際に、種々の記憶媒体を状況に応じて使い分けて利用することができる。

【0039】

また、請求項17にかかる発明は、請求項11～16のいずれか一つに記載のアプリ情報取得方法において、前記アプリ情報取得ステップによって取得するアプリ情報は、アプリケーションごとの固有の情報が含まれたプロダクトIDであることを特徴とする。

【0040】

この請求項17にかかる発明によれば、アプリ情報取得ステップが取得するアプリ情報として、アプリケーションごとの固有の情報を含むプロダクトIDを用いたため、アプリ情報のやり取りをこのプロダクトIDだけで、簡易かつ迅速に行うことができるとともに、アプリ情報が一元管理できることから、情報の信頼性を向上させることができる。

【0041】

また、請求項18にかかる発明は、請求項17に記載のアプリ情報取得方法において、前記プロダクトIDには、対応するアプリケーションのベンダー名とアプリケーション名に加えて、バージョン情報と使用リソース情報の少なくとも一方が含まれていることを特徴とする。

【0042】

この請求項18にかかる発明によれば、プロダクトIDには、対応するアプリケーションのベンダー名、アプリケーション名、および、バージョン情報と使用リソース情報の少なくとも一方が含まれているため、アプリ情報としてプロダクトIDを取得するだけで、アプリケーションの起動時に問題となるバージョンの整合性やリソース不足による動作の不安定要因の有無を正確に判断することができる。

【0043】

また、請求項19にかかる発明は、請求項17または18に記載のアプリ情報取得方法において、前記記憶ステップは、インストールを可能とするアプリケーションのプロダクトIDを記憶し、判断対象となるアプリケーションのプロダクトIDがその記憶ステップに記憶されているか否かによって、インストール可能なアプリケーションか否かを自動的に判断することを特徴とする。

【0044】

この請求項19にかかる発明によれば、記憶ステップでは、インストール可能なアプリケーションのプロダクトIDを記憶するようにし、判断対象のアプリケーションのプロダクトIDがその記憶ステップに記憶されているか否かを調べるだけで、そのアプリケーションがインストール可能か否かを自動的に判断できるようにしたため、簡易な構成だけで迅速かつ正確にインストール可能なアプリケーションか否かを自動判断することができる。

【0045】

【発明の実施の形態】

以下に添付図面を参照して、この発明にかかる画像形成装置およびアプリ情報取得方法の好適な実施の形態を詳細に説明する。

【0046】

図1は、この発明の本実施の形態である画像形成装置（以下、「複合機」という）の構成を示すブロック図である。図1に示すように、複合機100は、白黒レーザプリンタ（B&W LP）101と、カラーレーザプリンタ（Color LP）102と、スキャナ、ファクシミリ、ハードディスク、メモリ、ネットワークインターフェースなどのハードウェアリソース103を有するとともに、プラットホーム120とアプリケーション（以下、アプリともいう）130と仮想アプリケーションサービス（VAS：Virtual Application Service）140から構成されるソフトウェア群110とを備えている。

【0047】

また、本実施の形態では、アプリ130とプラットホーム120との間に仮想アプリケーションサービス（VAS）140を配置している。この仮想アプリケーションサービス（VAS）140は、コントロールサービスをサーバとしたクライアントプロセスとして動作し、かつアプリケーションをクライアントとしたサーバプロセスとして動作するもので、本発明の特徴的な構成要素であるアプリ情報取得手段としてのアプリ情報取得スレッドがこの仮想アプリケーションサービス140に含まれている。このVAS140は、あるアプリを起動可能な状態とするインストール処理を行う前に、アプリ情報を取得して、そのアプリ情報ファイルを記憶手段としてのハードディスク（HD）200に生成する。取得するアプリ情報の例としては、ここでは、プロダクト情報を取得するようにしていて、プロダクトID、ベンダー名、アプリ名、バージョン、連絡先（電話番号）、必要リソース、インストール時判定、関連アプリ情報などが含まれている。このプロダクト情報は、アプリ自身が持っていたり、ネットワーク経由で取得したり、あるいは、以前に取得したプロダクト情報をハードディスク200、不揮発性メモリであるNVRAM104、アプリ専用のアプリ情報テーブル106などに格納されている場合は、そのプロダクト情報を読み出すことによっても利用することができる。その際、VAS140は、インストール前のアプリに対して仮起動することにより、VAS140とアプリとの間だけでプロセス間通信により、アプリ情報を個別に取得することが可能となる。

【0048】

また、このアプリ情報をV A S 1 4 0 が取得する方法としては、アプリ自身は複合機100のハードディスク200にコピーしたり、P C カード、I C カード、S D カードといった種々のデータカードを介してコピーされた後、起動できるような状態とするインストール処理が行われることになる。本発明では、インストール処理を行う前に、アプリに関する情報であるアプリ情報を取得し、このアプリ情報に基づいてインストール処理が可能なアプリか否かを判断するようとする。このインストール処理が可能か否かの判断は、インストール処理後、そのアプリを起動させた場合に、インストールした複合機の既存のアプリや他のインストール済みの新規アプリとの関係においても、リソース不足やバージョンの整合性等の問題で複合機自体の動作が不安定となったり、動作が停止するおそれのない場合か否かによる。これは、起動できないアプリをインストールしても意味がない上、インストールされていると、誤って起動させた場合に複合機自体の動作に支障が出るからである。

【0049】

プラットホーム120は、アプリケーションからの処理要求を解釈してハードウェア資源の獲得要求を発生させるコントロールサービスと、一または複数のハードウェア資源の管理を行い、コントロールサービスからの獲得要求を調停するシステムリソーススマネージャ (S R M) 123と、汎用O S 121とを有している。

【0050】

コントロールサービスは、複数のサービスモジュールから形成され、S C S (システムコントロールサービス) 122と、E C S (エンジンコントロールサービス) 124と、M C S (メモリコントロールサービス) 125と、O C S (オペレーションパネルコントロールサービス) 126と、F C S (ファックスコントロールサービス) 127と、N C S (ネットワークコントロールサービス) 128とから構成されている。なお、このプラットホーム120は、あらかじめ定義された関数により前記アプリケーション130から処理要求を受信可能とするアプリケーションプログラムインタフェース (A P I) を有している。

【0051】

汎用OS121は、UNIX（登録商標）などの汎用オペレーティングシステムであり、プラットホーム120並びにアプリケーション130の各ソフトウェアをそれぞれプロセスとして並列実行する。

【0052】

SRM123のプロセスは、SCS122とともにシステムの制御およびリソースの管理を行うものである。SRM123のプロセスは、スキャナ部やプリンタ部などのエンジン、メモリ、HDDファイル、ホストI/O（セントロI/F、ネットワークI/F、IEEE1394 I/F、RS232C I/Fなど）のハードウェア資源を利用する上位層からの要求にしたがって調停を行い、実行制御する。

【0053】

具体的には、このSRM123は、要求されたハードウェア資源が利用可能であるか（他の要求により利用されていないかどうか）を判断し、利用可能であれば要求されたハードウェア資源が利用可能である旨を上位層に伝える。また、SRM123は、上位層からの要求に対してハードウェア資源の利用スケジューリングを行い、要求内容（例えば、プリンタエンジンにより紙搬送と作像動作、メモリ確保、ファイル生成など）を直接実施している。

【0054】

SCS122のプロセスは、アプリ管理、操作部制御、システム画面表示、LED表示、リソース管理、割り込みアプリ制御などを行う。

【0055】

ECS124のプロセスは、白黒レーザプリンタ（B&W LP）101、カラーレーザプリンタ（Color LP）102、スキャナ、ファクシミリなどからなるハードウェアリソースのエンジンの制御が行われる。

【0056】

MCS125のプロセスは、画像メモリの取得および解放、ハードディスク装置（HDD）の利用、画像データの圧縮および伸張などを行う。

【0057】

FCS127のプロセスは、システムコントローラの各アプリ層からPSTN／ISDN網を利用したファクシミリ送受信、BKM（バックアップSRAM）で管理されている各種ファクシミリデータの登録／引用、ファクシミリ読みとり、ファクシミリ受信印刷、融合送受信を行うためのAPIを提供する。

【0058】

NCS128のプロセスは、ネットワークI/Oを必要とするアプリケーションに対して共通に利用できるサービスを提供するためのプロセスであり、ネットワーク側から各プロトコルによって受信したデータを各アプリケーションに振り分けたり、アプリケーションからデータをネットワーク側に送信する際の仲介を行う。具体的には、ftp、http、lpd、snmpd、telnetd、smtpなどのサーバーデーモンや、同プロトコルのクライアント機能などを有している。

【0059】

OCS126のプロセスは、オペレータ（ユーザ）と本体制御間の情報伝達手段となるオペレーションパネル（操作パネル）105の制御を行う。OCS126は、オペレーションパネル105からキー押下をキーイベントとして取得し、取得したキーに対応したキーイベント関数をSCS122に送信するOCSプロセスの部分と、アプリケーション130またはコントロールサービスからの要求によりオペレーションパネルに各種画面を描画出力する描画関数やその他オペレーションパネルに対する制御を行う関数などがあらかじめ登録されたOCSライブラリの部分とから構成される。このOCSライブラリは、アプリケーション130およびコントロールサービスの各モジュールにリンクされて実装されている。なお、OCS126のすべてをプロセスとして動作させるように構成しても良く、あるいはOCS126のすべてをOCSライブラリとして構成しても良い。

【0060】

アプリケーション130は、ページ記述言語（PDL）、PCLおよびポストスクリプト（PS）を有するプリンタ用のアプリケーションであるプリンタアプリ111と、コピー用アプリケーションであるコピーアプリ112と、ファクシミリ用アプリケーションであるファックスアプリ113と、スキヤナ用アプリケ

ーションであるスキャナアプリ114と、ネットワークファイル用アプリケーションであるネットファイルアプリ115と、工程検査用アプリケーションである工程検査アプリ116とを有している。これらの各アプリは、複合機特有（画像形成装置特有）のアプリであり、起動時にVAS140に対して自プロセスのプロセスIDとともにアプリ登録要求メッセージを送信し、アプリ登録要求メッセージを受信したVAS140によって、起動したアプリに対する登録処理が行われるようになっている。

【0061】

アプリケーション130の各プロセス、コントロールサービスの各プロセスは、関数呼び出しとその戻り値送信およびメッセージの送受信によってプロセス間通信を行いながら、コピー、プリンタ、スキャナ、ファクシミリなどの画像形成処理にかかるユーザサービスを実現している。

【0062】

このように、本実施の形態にかかる複合機100には、複数のアプリケーション130および複数のコントロールサービスが存在し、いずれもプロセスとして動作している。そして、これらの各プロセス内部には、一または複数のスレッドが生成されて、スレッド単位の並列実行が行われる。そして、コントロールサービスがアプリケーション130に対し共通サービスを提供しており、このため、これらの多数のプロセスが並列動作、およびスレッドの並列動作を行って互いにプロセス間通信を行って協調動作をしながら、コピー、プリンタ、スキャナ、ファクシミリなどの画像形成処理にかかるユーザサービスを提供するようになっている。また、複合機100には、サードベンダなどの第三者がコントロールサービス層の上のアプリケーション層に新規アプリ117、118を開発して搭載することが可能となっている。図1では、この新規アプリ117、118を搭載した例を示している。

【0063】

なお、本実施の形態にかかる複合機100では、複数のアプリケーション130のプロセスと複数のコントロールサービスのプロセスとが動作しているが、アプリケーション130とコントロールサービスのプロセスがそれぞれ単一の構成

とすることも可能である。また、各アプリケーション130は、アプリケーションごとに追加または削除することができる。

【0064】

図2は、本実施の形態にかかる複合機100のVAS140の構成と、VAS140と各アプリ、コントロールサービス層150および汎用OS121との関係を示すブロック図である。なお、図2では、アプリケーション130の例として、プリントアプリ111、コピーアプリ112、新規アプリ117, 118を示しているが、他のアプリでも同様の構成である。

【0065】

仮想アプリケーションサービス(VAS)140のプロセスには、ディスパッチャ145と、制御スレッド144と、アプリ情報取得手段としてのアプリ情報取得スレッド141と、オペレーションパネルに対して操作画面を生成する画面表示スレッド142、および、取得したアプリ情報に基づいてアプリのインストールが可能か否かをチェックするアプリチェックスレッド143が動作している。

【0066】

ディスパッチャ145は、アプリケーション130やコントロールサービスからのメッセージ受信を監視し、受信したメッセージに応じて制御スレッド144、アプリ情報取得スレッド141、画面表示スレッド142、アプリチェックスレッド143に対して処理要求を行うものである。本実施の形態の複合機100では、ディスパッチャ145は、コントロールサービスからインストールされているアプリに対するアプリ起動要求メッセージを受信したとき、受信したアプリ起動要求メッセージを制御スレッド144に送信し、制御スレッド144からアプリ情報取得スレッド141、画面表示スレッド142、アプリチェックスレッド143に対して処理要求を行うようになっている。

【0067】

制御スレッド144は、ディスパッチャ144からのアプリ起動要求メッセージを受信して、アプリ情報取得処理、画面表示処理、およびアプリチェック処理を行う。ここで、アプリチェック処理とは、インストール処理を行うアプリがイ

ンストール後に起動させたとしても、複合機に対して影響を与える可能性が有るか無いかをチェックする処理のことである。例えば、インストール要求のあったアプリと、他のインストール済みのアプリのアプリ情報を取得して、バージョンチェックやリソースチェックを行うことによって、インストール後に起動させても動作に影響を与えないか否かを判断することが可能となる。その際、画面表示スレッド142は、アプリ情報取得スレッド141によって取得され、ハードディスク（HD）200の中のアプリ情報ファイル201に格納したアプリ情報を用いてオペレーションパネル105上に各アプリのアプリ情報を表示させることにより、インストールしようとしているアプリの状況を把握することができるので、アプリチェックスレッド143によってインストールが不可能であると判断されても、どのような理由でインストールできなかったのかを理解することができる。

【0068】

また、制御スレッド143は、アプリチェックスレッド143によってインストールが可能と判断されたり、あるいは、後述するように、インストールされているアプリが不要になった場合などに削除するアンインストール処理などを制御する。

【0069】

アプリ情報取得スレッド141は、制御スレッド143からの処理要求を受けると、インストール前のアプリに対しては、仮起動を行うことによってプロセス間通信が可能な状態として、アプリ情報を取得する。また、インストールされているアプリは、起動させることで同様にプロセス間通信によるアプリ情報取得が可能となる。このようにして、取得された各アプリのアプリ情報は、アプリ情報ファイル201としてハードディスク（HD）200に生成される。かかるアプリ情報は、アプリごとのレコードとして記録される。

【0070】

上記VAS140のプログラムは、ソフトウェア開発キット（SDK：Software Development Kit）等の一部または全部として、CD-ROMまたはFD（フレキシブルディスク）などの記憶媒体に実行可能な形

式またはインストール可能な形式のファイルで提供される。また、このような実行可能な形式またはインストール可能な形式のV A S 1 4 0 のプログラムファイルを、ネットワーク経由で取得可能な方法で提供するようにしても良い。

【0071】

図3は、V A S 1 4 0 とアプリ情報を格納するハードディスク200、アプリ情報テーブル106、N V R A M 1 0 4との関係を示す説明図である。図3に示すように、ハードディスク200に格納されているプロダクト情報は、各アプリ単位ごとに管理され、プロダクトIDが特定できれば、これに対応するアプリ情報も特定することができる。また、このハードディスク200には、複合機にコピーされたインストール前のアプリAやアプリCのデータが格納されており、また、アプリBは、I Cカードなどに格納されたデータを使ってインストール処理を行うものである。図3中の破線で示したFは、特定のパスにある実行ファイル群を示している。

【0072】

N V R A M 1 0 4 は、電源を切ってもメモリ内容を保持する不揮発性メモリであり、アプリA、アプリBのように、各アプリごとにプロダクトID、起動情報、付加情報がそれぞれ格納されている。

【0073】

また、アプリ情報テーブルは、アプリに関する情報を各アプリごとにテーブルデータとして格納したものである。このため、アプリAのプロダクトIDを入力すれば、このアプリAのアプリ名、バージョン、インストール属性等のアプリ情報を取り出すことができる。

【0074】

図4は、本実施の形態の複合機においてアプリに対してインストール処理かアンインストール処理を行う場合の全体フローチャートであり、図5は、図4のステップS401におけるアプリ情報取得処理のサブルーチンを示すフローチャートである。

【0075】

まず、図4のステップS401では、ディスパッチャ145が制御スレッド1

44を介してアプリ情報スレッド141に対して、インストール前、あるいは、インストール済みのアプリに対するアプリ情報取得処理を要求する。アプリ情報スレッド141は、後述する図5におけるアプリ情報取得処理を行って、アプリ情報ファイル201を生成する。

【0076】

また、上記ステップS401では、生成されたアプリ情報ファイル201から制御スレッド144がアプリ情報を読み出して、オペレーションパネル105に表示する操作画面を生成して、画面表示が行われる（ステップS402）。この表示画面は、コピーされたアプリに関するアプリ名、バージョン、リソース等の情報を併記して表示するとともに、選択画面に沿ってユーザが選択していくだけで、インストールの可否やインストール不可能になった場合の状況等が画面表示により理解できるようになっているので、ユーザは、その画面表示に従って選択処理を行う。

【0077】

全ての項目についてユーザの選択処理が行われた後は、ステップS403において、ユーザ操作待機の後（ステップS403）、ユーザ要求の有無が判断される（ステップS404）。ここでは、ユーザの要求として、アプリのインストール要求か、アンインストール要求のいずれかである。ユーザ要求がインストールで、許可されるならば、図3に示すNVRAM104に対して登録処理が行われる（ステップS405）。また、ユーザ要求がアンインストールであるならば、図3に示すNVRAM104からアプリ情報を削除することによってアンインストールされることになる（ステップS406）。

【0078】

図5は、図4におけるアプリ情報取得処理動作を詳細に説明したものである。VAS140は、インストール前あるいは、起動前のアプリに対してはそのままではアプリ情報を取得することができないため、情報取得対象のアプリに対して仮起動させ（ステップS501）、プロセス間通信により、所望の情報を要求することでアプリ情報を取得することができる。本実施の形態では、アプリ情報の一元管理が可能なプロダクトIDを要求してこれを取得すると（ステップS50

2)、NVRAM104を参照することにより、登録済みのアプリ情報か否かがわかる（ステップS503）。

【0079】

NVRAM104に登録済みの場合は、プロダクト情報ファイルの有無が判断され（ステップS504）、プロダクト情報がある場合は、アプリよりプロセス間通信を使って各種情報を取得するようとする（ステップS505）。これにより、プロダクト情報ファイルを生成した後（ステップS506）、インストール判定処理が行われる（ステップS507）。そして、その判定処理結果をプロダクト情報に反映させるようとする（ステップS508）。

【0080】

そして、ステップS509では、その判定結果に基づいて、インストールの可否が判断される。また、上記ステップS504で、プロダクト情報ファイルが有る場合は、プロダクト情報よりアプリ情報取得があつて（ステップS510）、インストール時の判定情報取得が行われた後（ステップS511）、上記したステップS509において、インストールの可否が判断される。

【0081】

ステップS509において、インストール可能であれば、インストール可能フラグの設定を行い（ステップS512）、アプリ情報テーブルへの書き込みが行われる（ステップS516）。また、インストールが不可の場合は、インストール不可フラグの設定が行われて（ステップS513）、アプリ情報テーブルへの書き込みが行われる（ステップS516）。さらに、上記したステップS503において、NVRAM104にプロダクトIDが登録されていないアプリの場合は、NVRAM104よりアプリ情報取得が行われ（ステップS514）、インストール済みのフラグの設定が行われて（ステップS515）、アプリ情報テーブルへの書き込みが行われる（ステップS516）。

【0082】

そして、全アプリに対する処理が終了していなければ、再び上記ステップS501に戻って、全アプリに対する処理が終了するまで、上記処理が繰り返される。

【0083】

このように、本実施の形態の複合機100では、新規アプリをインストール処理をする場合に、仮想アプリケーションサービス140のアプリ情報取得スレッド141を使って、全アプリに対してアプリ情報の取得が行われるため、この取得したアプリ情報に基づいて、インストールした後に当該アプリを起動した場合の動作状況を正確に把握することが可能となり、インストール後に起動させた場合でも、複合機の動作が不安定になったり、動作が停止するような状況を防止することが可能となる。

【0084】

【発明の効果】

以上説明したように、請求項1にかかる発明によれば、画像形成装置にコピーされたアプリケーションを起動可能な状態にするインストール処理を行う前に、アプリ情報取得手段によって、インストールが可能なアプリケーションか否かを判断するためのアプリケーションに関するアプリ情報の取得を行うようにしたので、インストール前でもインストールが可能なアプリケーションか否かを正確な判断を下すことができる。

【0085】

また、請求項2にかかる発明によれば、画像形成処理で使用されるハードウェア資源と、画像形成処理にかかるユーザサービスにそれぞれ固有の処理を行うアプリケーションと、前記アプリケーションと前記ハードウェア資源との間に介在し、ユーザサービスを提供する際に、アプリケーションの少なくとも2つが共通的に必要とするハードウェア資源の獲得要求、管理、実行制御並びに画像形成処理を行うコントロールサービスとをさらに備えているので、複数のアプリケーションを作成する際に、共通化できる部分をコントロールサービス（プラットホーム）として共通化することより、それ以外の部分を作成すればよいことから、アプリケーションの開発が容易となり、かつ、短期間で開発することができる。

【0086】

また、請求項3にかかる発明によれば、画像形成装置にコピーされたインストール処理前のアプリケーションが自らアプリ情報を保有している場合、アプリ情

報取得手段は、このアプリが保有するアプリ情報を取得するようにしたので、アプリケーションから直接正確な情報を取得することができる。

【0087】

また、請求項4にかかる発明によれば、画像形成装置にコピーされたアプリケーションインストール処理前のアプリケーションを仮起動することで、アプリ情報取得手段は、アプリケーションのアプリ情報を取得するようにしたので、通常起動前のアプリケーションに対してもプロセス間通信を用いて情報をやり取りすることにより、正確なアプリ情報を取得することができる。

【0088】

また、請求項5にかかる発明によれば、アプリ情報取得手段は、ネットワーク経由で画像形成装置にコピーされたインストール処理前のアプリケーションのアプリ情報を取得するようにしたので、常に最新の情報と更新可能なネットワークデータを使って正確なアプリ情報を取得することができる。

【0089】

また、請求項6にかかる発明によれば、アプリ情報取得手段が取得したアプリ情報を記憶手段に記憶するようにしたので、一旦取得したアプリ情報を必要に応じていつでも取り出せることから、データベースのように使用することが可能となり、アプリ情報取得に要する時間を短縮して、処理の迅速化が図れる。

【0090】

また、請求項7にかかる発明によれば、ハードディスク、不揮発性メモリ、および各種カードの少なくとも一つから記憶手段が構成され、アプリ情報取得手段によって、それらの記憶媒体に記憶された情報を読み出すことでアプリ情報を取得するようにしたので、種々の記憶媒体をアプリ情報の記憶手段として利用することができとなり、状況に応じて記憶媒体を使い分けることができる。

【0091】

また、請求項8にかかる発明によれば、アプリ情報取得手段が取得するアプリ情報として、アプリケーションごとの固有の情報を含むプロダクトIDを用いたので、アプリ情報のやり取りをこのプロダクトIDだけで、簡易かつ迅速に行うことができるとともに、アプリ情報が一元管理できることから、情報の信頼性を

向上させることができる。

【0092】

また、請求項9にかかる発明によれば、プロダクトIDには、対応するアプリケーションのベンダー名、アプリケーション名、および、バージョン情報と使用リソース情報の少なくとも一方が含まれているので、アプリ情報としてプロダクトIDを取得するだけで、アプリケーションの起動時に問題となるバージョンの整合性やリソース不足による動作の不安定要因の有無を正確に判断することができる。

【0093】

また、請求項10にかかる発明によれば、記憶手段には、インストール可能なアプリケーションのプロダクトIDを記憶するようにし、判断対象のアプリケーションのプロダクトIDがその記憶手段に記憶されているか否かを調べるだけで、そのアプリケーションがインストール可能か否かを自動的に判断できるようにしたので、簡易な構成だけで迅速かつ正確にインストール可能なアプリケーションか否かを自動判断することができる。

【0094】

また、請求項11にかかる発明によれば、画像形成装置にコピーされたアプリケーションを起動可能な状態にするインストール処理を行う前に、アプリ情報取得ステップによって、インストールが可能なアプリケーションか否かを判断するためのアプリケーションに関するアプリ情報の取得を行うようにしたので、インストール前でもインストールが可能なアプリケーションか否かを正確な判断を下すことができる。

【0095】

また、請求項12にかかる発明によれば、画像形成装置にコピーされたインストール処理前のアプリケーションが自らアプリ情報を保有している場合、アプリ情報取得ステップは、このアプリケーションが保有するアプリ情報を取得するようにしたので、アプリケーションから直接正確な情報を取得することができる。

【0096】

また、請求項13にかかる発明によれば、画像形成装置にコピーされたアプリ

ケーションインストール処理前のアプリケーションを仮起動することで、アプリ情報取得ステップによって、アプリケーションのアプリ情報を取得するようにしたので、通常起動前のアプリケーションに対してもプロセス間通信を用いて情報をやり取りすることが可能となり、正確なアプリ情報を取得することができる。

【0097】

また、請求項14にかかる発明によれば、アプリ情報取得ステップは、ネットワーク経由で画像形成装置にコピーされたインストール処理前のアプリケーションのアプリ情報を取得するようにしたので、常に最新の情報と更新可能なネットワークデータを使って正確なアプリ情報を取得することができる。

【0098】

また、請求項15にかかる発明によれば、アプリ情報取得ステップによって取得したアプリ情報を記憶ステップにて記憶するようにしたので、一旦取得したアプリ情報を必要に応じていつでも取り出すことができ、データベースのような使用が可能となり、アプリ情報取得に要する時間を短縮して、処理の迅速化が図れる。

【0099】

また、請求項16にかかる発明によれば、アプリ情報取得ステップは、記憶ステップによってハードディスク、不揮発性メモリ、および各種カードの少なくとも一つの記憶媒体に記憶された情報を読み出すことでアプリ情報が取得できるようにしたので、アプリ情報を記憶しておく際に、種々の記憶媒体を状況に応じて使い分けて利用することができる。

【0100】

また、請求項17にかかる発明によれば、アプリ情報取得ステップが取得するアプリ情報として、アプリケーションごとの固有の情報を含むプロダクトIDを用いたので、アプリ情報のやり取りをこのプロダクトIDだけで、簡易かつ迅速に行うことができるとともに、アプリ情報が一元管理できることから、情報の信頼性を向上させることができる。

【0101】

また、請求項18にかかる発明によれば、プロダクトIDには、対応するアプ

リケーションのベンダー名、アプリケーション名、および、バージョン情報と使用リソース情報の少なくとも一方が含まれているので、アプリ情報としてプロダクトIDを取得するだけで、アプリケーションの起動時に問題となるバージョンの整合性やリソース不足による動作の不安定要因の有無を正確に判断することができる。

【0102】

また、請求項19にかかる発明によれば、記憶ステップでは、インストール可能なアプリケーションのプロダクトIDを記憶するようにし、判断対象のアプリケーションのプロダクトIDがその記憶ステップに記憶されているか否かを調べるだけで、そのアプリケーションがインストール可能か否かを自動的に判断できるようにしたので、簡易な構成だけで迅速かつ正確にインストール可能なアプリケーションか否かを自動判断することができる。

【図面の簡単な説明】

【図1】

実施の形態1にかかる複合機の構成を示すブロック図である。

【図2】

本実施の形態にかかる複合機のVASの構成と、VASと各アプリ、コントロールサービス層および汎用OSとの関係を示すブロック図である。

【図3】

VASとアプリ情報を格納するハードディスク、アプリ情報テーブル、NVRAMとの関係を示す説明図である。

【図4】

本実施の形態の複合機においてアプリに対してインストール処理かアンインストール処理を行う場合の全体フローチャートである。

【図5】

図4のステップS401におけるアプリ情報取得処理のサブルーチンを示すフローチャートである。

【符号の説明】

100 複合機

- 101 白黒レーザプリンタ
- 102 カラーレーザプリンタ
- 104 N V R A M
- 105 オペレーションパネル
- 110 ソフトウェア群
- 111 プリンタアプリ
- 112 コピーアプリ
- 113 ファックスアプリ
- 114 スキヤナアプリ
- 115 ネットファイルアプリ
- 116 工程検査アプリ
- 117, 118 新規アプリ
- 120 プラットホーム
- 121 汎用O S
- 122 S C S
- 123 S R M
- 124 E C S
- 125 M C S
- 126 O C S
- 127 F C S
- 128 N C S
- 130 アプリケーション
- 140, 841~848 仮想アプリケーションサービス (V A S)
- 141 アプリ情報取得スレッド
- 142 画面表示スレッド
- 143 アプリチェックスレッド
- 144 制御スレッド
- 145 ディスパッチャ
- 150 コントロールサービス層

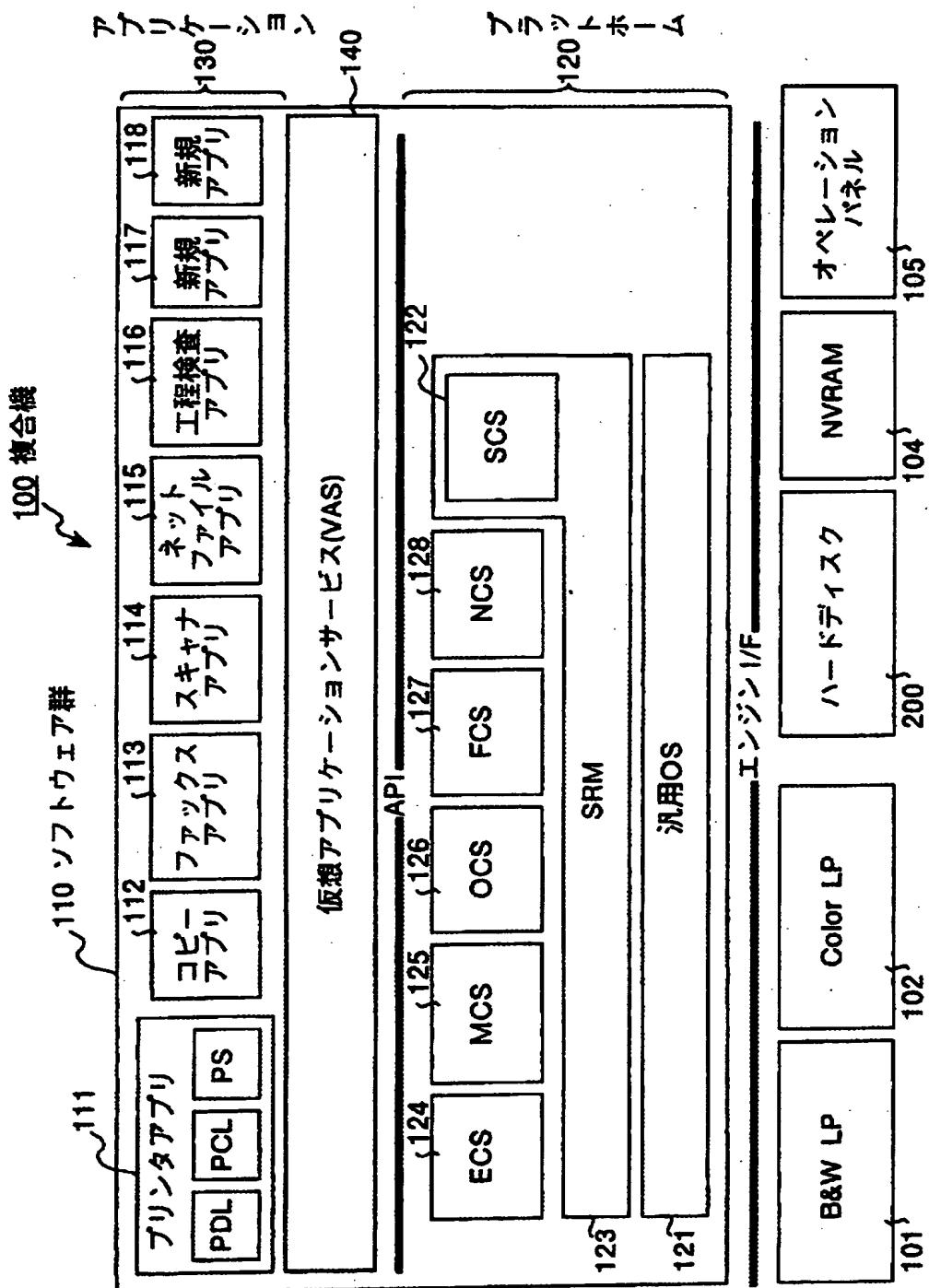
200 ハードディスク (H D)

201 アプリ情報ファイル

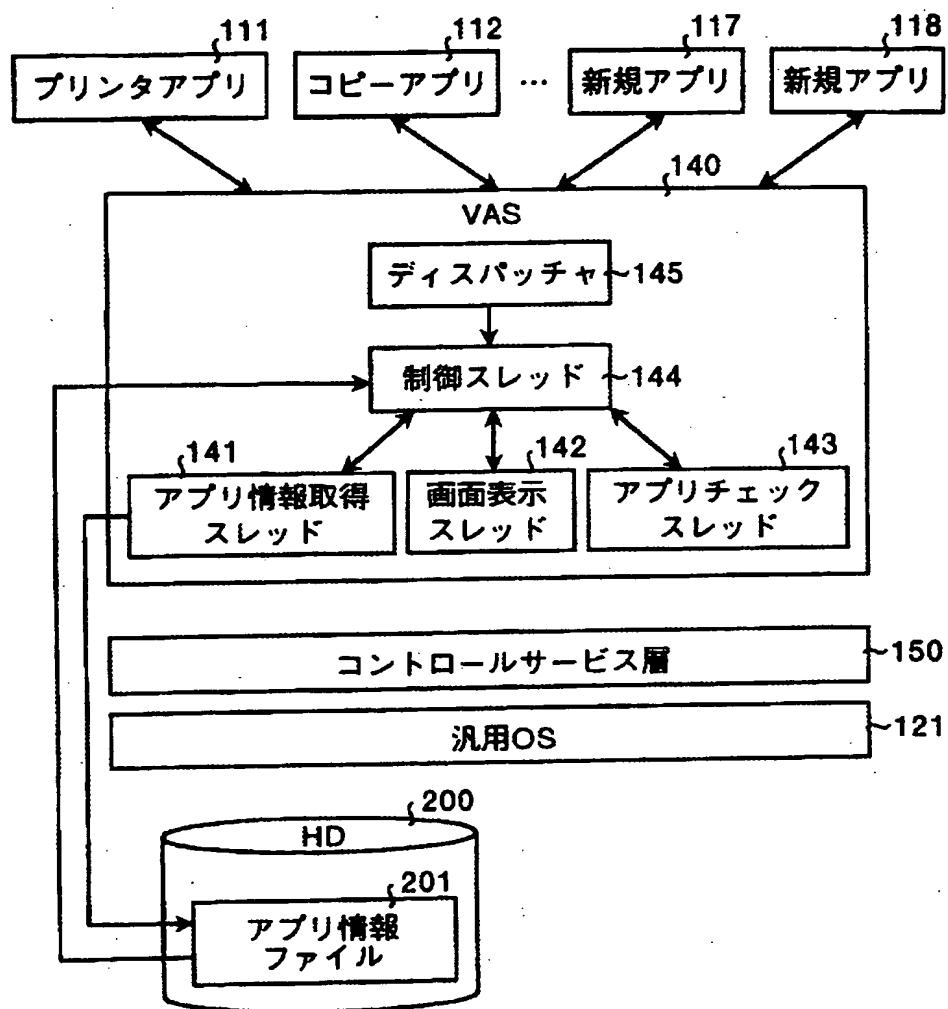
【書類名】

図面

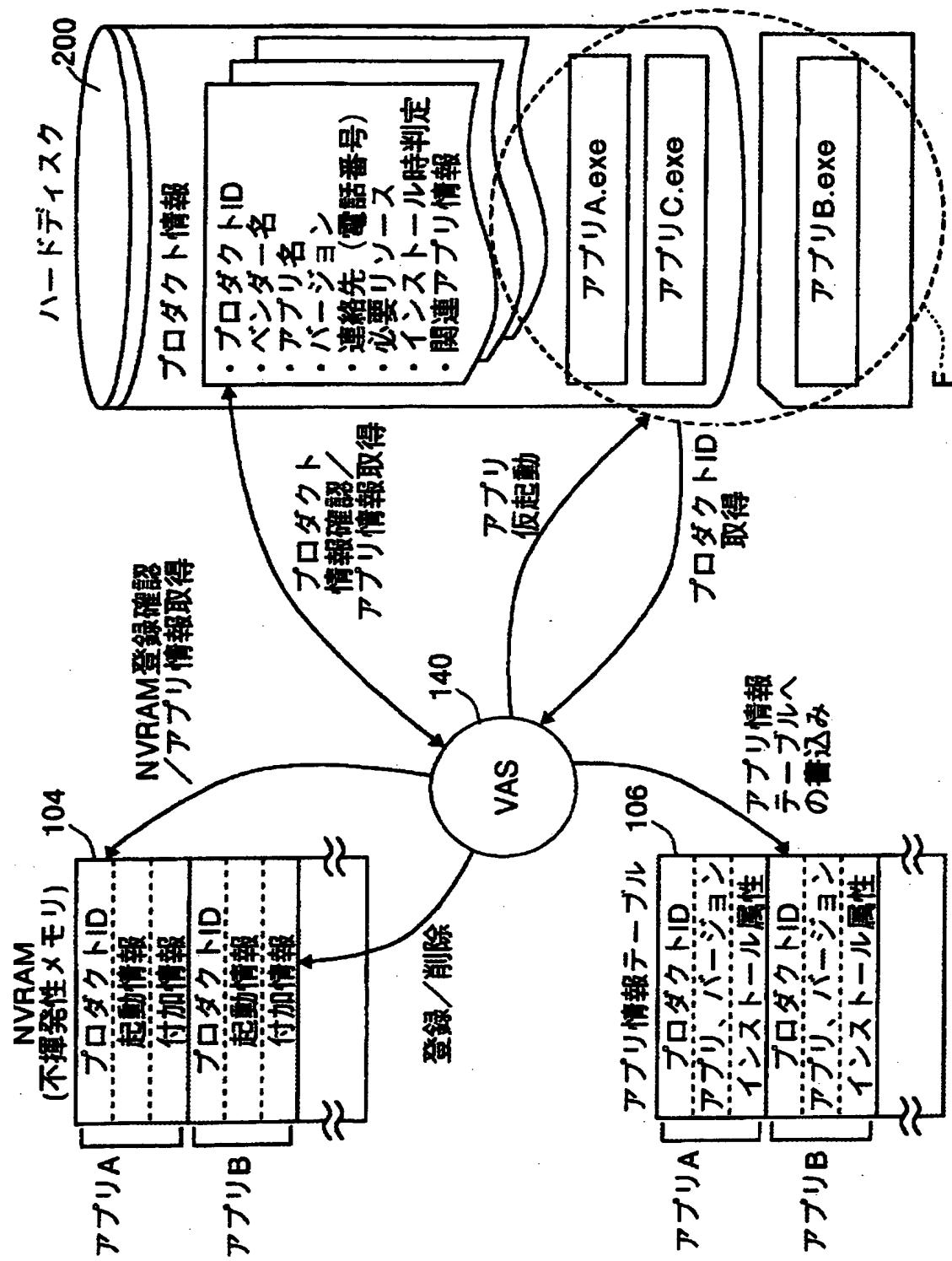
【図1】



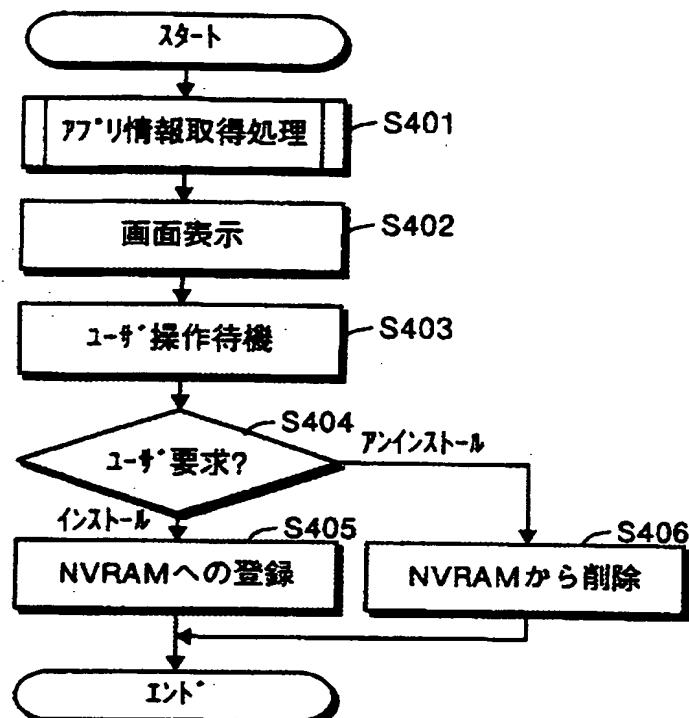
【図2】



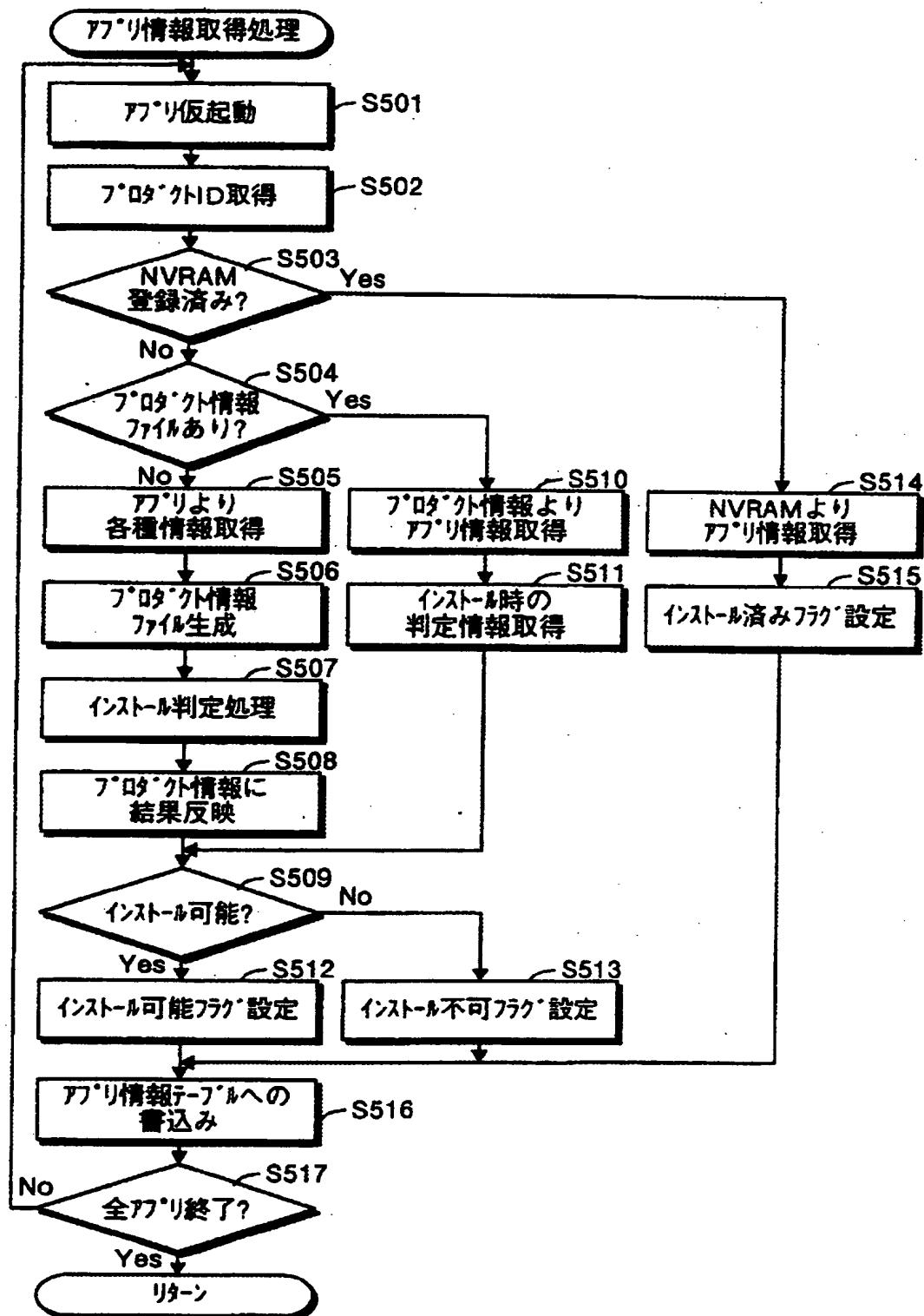
【図 3】



【図4】



【図5】



【書類名】 要約書

【要約】

【課題】 インストール処理の前に、インストールが可能なアプリケーションか否かを判断するために必要なアプリケーションに関する正確なアプリ情報取得することができる。

【解決手段】 新規アプリのインストール要求があると、アプリ情報取得スレッド141は、複合機の全アプリに対してアプリ情報の取得処理を行う。アプリ情報として、プロダクトIDを利用し、各アプリ単位でアプリ情報を管理するようする。NVRAMにプロダクトIDが登録されていれば、インストールされたアプリであり、登録されていなければ未インストールのアプリである。未インストールのアプリの場合は、プロダクト情報ファイルを生成して、インストールの判定処理をアプリチェックスレッドにより行うことで、インストールの可否を正確に判定することができる。

【選択図】 図2

認定・付加情報

特許出願の番号	特願2002-307637
受付番号	20201770086
書類名	特許願
担当官	田口 春良 1617
作成日	平成14年11月27日

<認定情報・付加情報>

【特許出願人】

【識別番号】	000006747
【住所又は居所】	東京都大田区中馬込1丁目3番6号
【氏名又は名称】	株式会社リコー

【代理人】

【識別番号】	100089118
【住所又は居所】	東京都千代田区霞が関3丁目2番6号 東京俱楽部ビルディング 酒井国際特許事務所
【氏名又は名称】	酒井 宏明

次頁無

特願 2002-307637

出願人履歴情報

識別番号 [000006747]

1. 変更年月日 1990年 8月24日
[変更理由] 新規登録
住 所 東京都大田区中馬込1丁目3番6号
氏 名 株式会社リコー

2. 変更年月日 2002年 5月17日
[変更理由] 住所変更
住 所 東京都大田区中馬込1丁目3番6号
氏 名 株式会社リコー